**CECS 329, LO9 Assessment, April 18th, Spring 2024, Dr. Ebert**

# Problems

LO5. Solve the following.

(a) Provide the instructions of a URM program that computes the predicate function $f(x, y) = (x > y)$. For example, $f(5, 3) = 1$, while $f(3, 3) = f(4, 7) = 0$.

**Solution.**

```
1:J(1,2,6)
2: J(1,3,6)
3: J(2,3,8)
4: S(3)
5: J(1,1,2)
6: Z(1)
7: J(1,1,10)
8: Z(1)
9: S(1)
```

(b) Describe the role each register plays in computing $f(x, y)$.

**Solution.** $R_1$ holds $x$, $R_2$ holds $y$, and $R_3$ counts upwards from 0. Assuming $x \neq y$, if $R_3$ first equals $R_2$, then $x > y$. Otherwise, $x < y$.

LO6. Solve the following problems.

(a) Provide the URM program $P$ whose Gödel number equals

$$2^{26} + 2^{48} + 2^{96} + 2^{113} - 1.$$

Show all work.

**Solution.** $\beta(I_1) = 26 \Rightarrow$ Transfer since $26 \bmod 4 = 2$

$(26-2)/4 = 6.$ $6+1=7 = 2^{\boxed{0}}((2)(\boxed{3})+1) \Rightarrow \boxed{T(1,4).}$

$\beta(I_2) = 48-26-1 = 21.$ $21 \bmod 4 = 1 \Rightarrow S(i)$

$(21-1)/4 = 5 \Rightarrow i = 6 \Rightarrow \boxed{S(6)}$

$\beta(I_3) = 96-48-1 = 47.$ $47 \bmod 4 = 3 \Rightarrow J(i,j,k).$

$(47-3)/4 = 11$ $11+1 = 12 = 2^{\boxed{2}}((2)(\boxed{1})+1)$

$2+1 = 3 = 2^{\boxed{0}}((2)(\boxed{1})+1) \Rightarrow \boxed{J(2,1,2)}$

$\beta(I_4) = 113-96-1 = 16$ $16/4 = 4 \Rightarrow Z(5).$

$\therefore P = T(1,4), S(6), J(2,1,2), Z(5)$

(b) A universal program $P_U$ is simulating a program that has 286 instructions and whose Gödel number is

$$x = 2^3 + 2^{179} + 2^{192} + 2^{215} + 2^{223} + 2^{275} + \cdots + 2^{c_{286}} - 1.$$

If the current configuration of the computation of $P_x$ on some input has encoding

$$\sigma = 2^2 + 2^6 + 2^8 + 2^{15} + 2^{19} - 1,$$

then provide the next configuration of the computation *and* its encoding.

**Solution.** $C = (2, 3, 1, 6, 3)$   $PC = 3$

$\beta(I_3) = 192 - 179 - 1 = 12$. $12/4 = 3 \Rightarrow Z(4)$

$C_{next} = (2, 3, 1, 0, 4)$   $\tau(C_{next}) = 2^2 + 2^6 + 2^8 + 2^9 + 2^{15} - 1$

LO7. Answer the following.

(a) Describe what it means to be a positive instance of the `Self Accept` decision problem.

**Solution.** See Section 5 of the Undecidability and the Diagonalization Method lecture.

(b) In applying the diagonalization method towards proving the undecidability of `Self Accept`, we defined a computable function $g(x)$ in terms of $f(x)$, where $f(x)$ is the decision function for `Self Accept`, which we assume to be total computable. Provide the formula for computing $g(x)$ and describe what needs to be established about $g(x)$ in order for the diagonalization method to be successful.

**Solution.** See Section 5 of the Undecidability and the Diagonalization Method lecture.

(c) Suppose URM program $P_{1269}$ computes $\phi_{1269}(x)$ and that

$$\phi_{1269}(x) = x \bmod 2.$$

What is the value of $g(1269)$ and how does this prove that $g \neq \phi(1269)$? Defend your answer

**Solution.** Since $\phi_{1269}(1269) = 1269 \bmod 2 = 1$, it follows that $g(1269) = \uparrow \neq \phi_{1269}(1269)$, and so $g \neq \phi_{1269}$.

LO8. An instance of the decision problem `Thousand-Plus Range` is a Gödel number $x$, and the problem is to decide if function $\phi_x$ has a non-empty range and whose range members have a value of at least 1000. Consider the function

$$g(x) = \begin{cases} 1 & \text{if } \phi_x \text{ has a thousand-plus range} \\ 0 & \text{otherwise} \end{cases}$$

(a) Evaluate $g(x)$ for each of the following Gödel number's $x$. Note: 2 out of 3 correct is considered passing. **Justify your answers**.

i. $x = e_1$, where $e_1$ is the Gödel number of the program that computes the function $\phi_{e_1}(y) = 4000y^2 + 300y$.
**Solution.** $g(e_1) = 0$ since $\phi_{e_1}(0) = 0 < 1000$.

ii. $x = e_2$, where $e_2$ is the Gödel number of the program that computes the function $\phi_{e_2}(y) = 1459$.
**Solution.** $g(e_2) = 1$ since $\phi_{e_2}(y) = 1459 > 1000$, for all $y \geq 0$.

iii. $x = e_3$, where $e_3$ is the Gödel number of the program that computes the function $\phi_{e_3}(y) = 2^{2^{4(y+1)}}$.
**Solution.** $g(e_3) = 1$ since $\phi_{e_3}(y)$ is an increasing function, and

$$\phi_{e_3}(0) = 2^{2^4} = 2^{16} > 2^{10} = 1024 > 1000.$$

(b) Prove that $g(x)$ is not URM computable. In other words, there is no URM program that, on input $x$, always halts and either outputs 1 or 0, depending on whether or not $\phi_x$ has a thousand-plus range. Do this by writing a program $P$ that uses $g$ and makes use of the `self` programming construct. Then use a proof by cases to show that $P$ creates a contradiction.

**Solution.** Program $P$ is shown below.

Input $y$.
If $g(\texttt{self}) = 1$, then return 0.
Else $//g(\texttt{self}) = 0$
    Return 1000.

Let $e$ denote the Gödel number of $P$.

**Case 1:** $g(e) = 1$. Then $P$ has the thousand-plus range property but outputs 0 for every input, a contradiction.

**Case 2:** $g(e) = 0$. Then $P$ does not have the thousand-plus range property but outputs 1000 for every input, a contradiction.

Therefore, `Thousand-Plus Range` must be undecidable since assuming otherwise leads to a contradiction.

LO9. Solve the following.

(a) Provide the state diagram for a DFA $M$ that accepts all binary words having an even number of 0's *and* does *not* contain the subword 101. Hint: make sure that, for each state of $M$, you are able to describe the current status of *both* properties in relation to being in that state.

(b) Show the computation of $M$ on inputs i) $w_1 = 1100100$ and ii) $w_2 = 0110100$.